UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

APPROXIMATION OF OVER-THE-LINE DELAY FOR SINGLE LINE - SINGLE SPEED FREIGHT RAIL TRANSPORTATION

by

Teodor Gabriel Crainic*+
and
Lydie Querin*

* Centre de recherche sur les transports, Université de Montréal
+ Département des sciences administratives, Université du Québec à Montréal

This research has been made possible due to a subvention from the International Development Research Centre of Ottawa.

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal
Centre de recherche sur les transports - Publication #519
May 1987

Dépôt légal - 2e trimestre 1987

ABSTRACT

Recently, we developed a multimode multiproduct network optimization model that is imbedded in an interactive-graphic system for the strategic analysis and planning of national freight transportation systems, called STAN. The model incorporates over-the-line delay functions that are used in the global objective to represent the impact of rail line capacity and congestion on the performance of the whole freight transportation system. The objective of this report is to present the formulae used to estimate the link train flows and the over-the-line delays in the rail model present in STAN, as well as the procedures that may be used for their calibration for the links of a large network.

KEY WORDS: RAIL FREIGHT TRANSPORTATION,
STRATEGIC PLANNING,
OVER-THE-LINE DELAY,
APPROXIMATION FUNCTIONS.

RESUME

Nous avons récement mis-au-point un modèle d'optimisation de réseau multimode multiproduit qui est incorporé au système interactif-graphic STAN pour l'analyse et la planification stratégiques des réseaux nationaux de transport de marchandises. Ce modèle fait appel aux fonctions de délai de parcours des voies pour refléter, au niveau de l'objectif global, l'impact de la capacité et de la congestion des lignes de chemin-de-fer sur tout le système multimodal de transport de marchandises. Ce rapport presente les formules utilisées dans ce modèle pour estimer les flux de trains ainsi que les délais de parcours des liens du sous-réseau ferrovaire, ainsi que les procedures permettant de calibrer ces fonctions pour le cas d'un réseau de taille réelle.

MOTS CLES: TRANSPORT FERROVIAIRE DE MARCHANDISES,
PLANIFICATION STRATEGIQUE,
DELAI DE PARCOURS DE VOIE,
FONCTIONS D'APPROXIMATION.

I. INTRODUCTION

In rail transportation, the train travel time between two consecutive stations depends not only on the distance to be travelled and on the average speed of the train, but also on the physical characteristics (signalization, number of sidings, single or double track, etc.) of the line and on its utilization as measured by the total number of trains that pass on the line during a given period of time.

In order to model this travel time, it is possible to develop *over-the-line delay* formulae that determine the *average train travel time* on a rail line segment as a function of the characteristics and *capacity* of the line and the *total train traffic*. Such formulae may be used at the operational level to forecast and analyse the performance of specific rail lines; similar but more aggregated formulae are a particularly important component of network optimization models of freight rail transportation systems at tactical (medium term) or strategic (long term) planning levels.

Recently, we developed a multimode multiproduct network optimization model (Florian et al. 1986a, Crainic, Florian and Guélat 1987a,b), that is imbedded in an interactive-graphic system for the strategic analysis and planning of national freight transportation systems, called STAN (Florian et al. 1986b). The model incorporates over-the-line delay functions that are used in the global objective function to represent the impact of rail line capacity and congestion on the performance of the whole freight transportation system.

Due to its multimode environment and strategic planning objectives, a number of approximations had to be made when building the network model. Among these, the most critical for an adequate representation of the rail mode are the estimation formulae for the total train traffic on each link and the polynomial approximation of the over-the-line delays.

The purpose of this report is to present these formulae, as well as procedures that may be used for their calibration, especially for the links of a large network. The reader interested in how the national rail model evolved and its integration into the STAN model may refer to Crainic, Florian and Leal (1987) or to Roy (1986).

We want to emphasize that the following discussion, procedures and general calibration approach are equally valid for all delay functions of the same general class and shape as those presented in this report. In particular, the methodology applies to the yard put-through delay functions described in Crainic, Florian and Leal (1987). We focus on over-the-line delay functions, however, because, by sheer numbers alone (a "few" yards compared to hundreds or even thousands of links in most realistic national applications), links and link characteristics play a central role in STAN and the strategic planning of national freight transportation systems.

The report is organized as follows. Section II presents the formulae used in the modeling of the rail mode, followed, in Section III, by the description of the data used for the experiments. Section IV is dedicated to the presentation and analysis of the calibration procedure for the polynomial over-the-line delay function. The link train flow approximation formula is then discussed in section V. The Appendix contains the computer codes and instructions for using the various procedures that may be helpful in performing the computations implied by the calibration of the delay functions for the links of a large network.

II. NOTATION AND BASIC FORMULAE

The general structure of the national freight network model and the requirements of the solution algorithm used place a number of restrictions and guide lines on the possible model formulations for each of the transportation modes that it considers, among which the most important are:

a) Only product flows, expressed in some measure of weight, may be defined on the network;
b) No explicit capacity constraints may be included in the model;
c) The total cost objective function must be convex.

For the rail mode, these hypotheses have two main consequences. First, as some important components of a rail transportation system with path-like structures (e.g. train services and frequencies, train and car flows) cannot be defined on the network, product flows must be transformed into an equivalent number of cars and trains for each link. These approximations are subsequently used in computing link delays and costs. Secondly, the link or over-the-line delays have to reflect the delays due to the train congestion and to enforce the capacity restrictions of the line.

A number of additional hypotheses were inspired by the context of the brazilian transportation system and by the needs of strategic planning:

a) Passenger trains are not considered on the network;
b) All freight trains have equal priority;
c) On a link, all trains moving in the same direction, travel at the same (and constant) speed;
d) Consequently, no more than two trains meet at the same time;
e) On each link, the traffic is balanced, meaning that the number of trains travelling in each direction is approximately the same.

A detailed presentation of the freight rail model based on these hypotheses may be found in Crainic, Florian and Leal (1987). Here, we only summarize the main results.

The number of trains travelling on a particular link a is computed by the following formula:

$$t_a = \sum_p \frac{(v_a^p + (v_a^p / w_p) * vw_p)}{(\alpha_p * \beta_a)}$$

where,

t_a : *number of trains on link a;*

v_a^p : *quantity of product p travelling on link a (weight units);*

w_p : *quantity of product p loaded into a vehicle p (weight units);*

vw_p : *weight of an empty vehicle p (weight units);*

α_p : *weight of a train of product p (weight units);*

β_a : *link correcting factor.*

The average over-the-line delay for a train travelling on a single track link may be computed, consistent with the assumptions made and given the capacity of the link and the train flows in each direction, by a formula proposed by Petersen (1974):

$$D_a = \frac{FRT_a * CAP_a + (FRT_{\bar{a}} * t_{\bar{a}} - FRT_a * t_a)}{CAP_a - (t_{\bar{a}} + t_a)}$$

where the two links **a** and $\bar{a}$ represent the two directions of travel on a single track rail line and where,

D_a : *delay on link a;*

CAP_a : *capacity of link a;*

FRT_a : *free running time on link a;*

$FRT_{\bar{a}}$: *free running time on link* $\bar{a}$;

t_a : *train flow on link a;*

$t_{\bar{a}}$: *train flow on link* $\bar{a}$.

We identified several problems related to the above formula:

a) The asymptotic (toward infinity) behavior of the delay when the total train flow approaches the link capacity (see figure 1);

b) When fully developed, by substituting the equation for the number of trains, the formula consists of too many terms, implying a more difficult user interface, as well as longer computation times for the function evaluation procedure of the assignment algorithm;

c) The convexity of an objective function that includes this delay formula cannot be demonstrated in a general case.

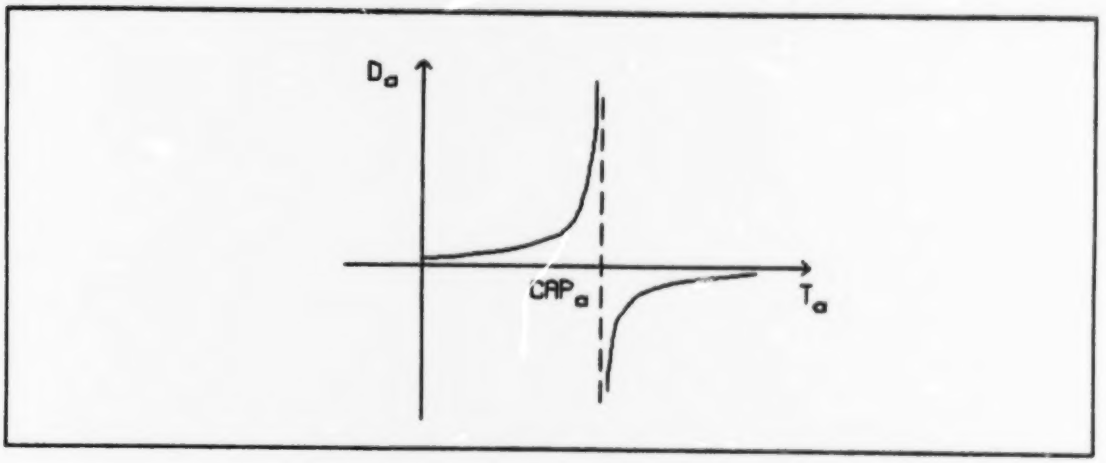

Figure 1. General shape of a delay function

We have therefore proposed a polynomial approximation of the delay function:

$$\overline{D}_a = FRT_a * (1 + \phi_1 * T_a + \phi_2 * (T_a / CAP'_a)^{exp})$$

where,

$\overline{D}_a$: *estimate (approximation) of delay on link a;*

T_a : *total train flow on the link*

$$T_a = t_a + t_{\bar{a}}$$

CAP'_a : *value of the total train flow where the line begins to be considered congested;*

ϕ_1 : *linear parameter (to be calibrated);*

ϕ_2 : *nonlinear (congestion) parameter (to be calibrated);*

exp : *linear parameter (to be calibrated).*

The parameters may be calibrated by non-linear regression methods, and we present results of this procedure later on. Simply used however, this approach is not entirely satisfactory for real applications, since too large a number of different polynomials would be generated (in fact, a different polynomial for each link of the network!). Also, the degree of the polynomials generated by non-linear regression methods are, generally, quite high, increasing the solution time of the assignment algorithm.

As an alternative, a heuristic procedure may be used to calibrate the parameters of the polynomial approximation. The specific objective of this paper is to present the details of this heuristic approach, to examine its efficiency and goodness of fit and to explore its utilization to generate a limited number of "accurate" delay formulas for the links of a large network. First, we briefly present the data used for the experiments.

III. DATA DESCRIPTION

Our experiments were conducted with data from the Canadian National Railways (CNR). It consists of traffic demand forecasts for the western part of CNR's network. Once aggregated for tactical planning purposes, it represents a network of 107 nodes and 95 links of which 86 are single track links. Traffic demand is defined for 25 aggregated products, while two freight train types (normal and rapid), as well as passenger trains, are allowed on the network.

A tactical planning network model (Crainic, Ferland and Rousseau, 1984) has been applied with this data in order to obtain train service routings and frequencies, yard classification workloads and traffic itineraries through the service (trains and yards) network. Detailed over-the-line delay formulae are used in this model (Crainic, 1982).

For each single track link, the following data is available:

- origin and destination nodes;
- length;
- free running time;
- capacity (estimated) per week;
- number of passenger trains (both directions);
- traffic volumes per product (both directions);
- number of freight trains of each train type (both directions);
- average over-the-line delay for each train type.

A constant speed of 45 miles/h was assumed for all links, which simplified the delay formula to:

$$D_a = \frac{FRT_a * (CAP_a + t_{\bar{a}} - t_a)}{CAP_a - t_{\bar{a}} - t_a}$$

As the line capacities were not readily available, being implicitly considered by the over-the-line delay formula, Colson's formula was used to estimate the link capacity for this experiment:

$$CAP_a = \frac{K * (2 * 1440 * 7)}{S_a^2 + \Phi}$$

where,

S_a : *siding to siding time;*

Φ : *signalization time;*

K : *percentage of effective capacity, accounting for the maintenance of the line.*

We used two minutes signalization time, 85% effective capacity and a week as planning period.

A sample of 10 links was extracted from the network, with the objective of selecting various combinations of link capacity and length. Table 1 displays the sample and its characteristics.

link number	link	FRT	CAP
1	(5,15)	293,33	212,92
2	(17,18)	18,67	193,57
3	(19,23)	398,67	72,75
4	(24,31)	562,67	123,73
5	(26,31)	489,33	48,42
6	(45,46)	52,00	223,69
7	(49,50)	72,00	261,35
8	(59,65)	161,33	146,57
9	(70,73)	168,00	237,72
10	(104,105)	52,00	25,44

Table 1. Link sample

In an exploratory approach, that is as much art as it is science, our initial objective is to calibrate a polynomial delay function for each link in the sample. We then try to use these polynomials to generate one polynomial delay function which adequately fits all links or, at least, a few polynomials each of which adequately fits a subset of links.

IV. CALIBRATION PROCEDURE

This section is dedicated to the presentation of the calibration procedure for the parameters of the polynomial over-the-line delay function. It begins with a general statement of the procedure, followed by the presentation of the results of the numerical experiments performed using the sample data and the conclusions that we draw concerning the calibration of each one of the parameters. Based on these conclusions, we suggest possible approaches for the efficient calibration of a limited number of polynomial delay functions for the links of a realistically large network. The last part of this section presents a validation of the polynomial functions calibrated by our procedure by comparing their goodness of fit with that of the polynomials obtained by non-linear regresssion methods.

Note that when the appropriate function is introduced in its first step, the calibration procedure may be used to approximate any delay function of the general form illustrated in Figure 2, inclusive, in particular, of yard delay functions.

4.1 The procedure

The procedure estimates values for the three parameters, exp, ϕ_1 and ϕ_2, of the polynomial delay function $\overline{D}_a(\phi_1, \phi_2, \text{exp}, T_a)$, such that it fits closely to the delay $D_a(t_a, t_{\bar{a}})$ computed by a given formula. Here, Peterson's delay (1974) is used.

For link **a**, the general procedure is:

Step 1. Generate a set of points (T_a, D_a) = (total train traffic, delay), $T_a = t_a + t_{\bar{a}}$, for various combinations of train flows t_a and $t_{\bar{a}}$. Fix the value of CAP'_a at the level of the total train flow where the link begins to be considered congested.

Step 2. Plot the points obtained in Step 1, "observe" the curvature of the scatter diagram and fix **exp** accordingly.

Step 3. Estimate ϕ_2 by fitting

$$D_a = FRT_a * (1 + \phi_2 * (T_a / CAP'_a)^{\text{exp}})$$

for high values of T_a, i.e. $T_a \geq CAP'_a$.

Step 4. Evaluate ϕ_1 to adjust

$$D_a = FRT_a * (1 + \phi_1 * T_a^* + \phi_2 * (T_a^* / CAP'_a)^{exp})$$

for a point in the linear part of the scatter diagram, e.g. $T_a^* \in [0{,}1 * CAP_a,\ 0{,}2 * CAP_a]$, and the value of ϕ_2 calculated at Step 3.

Step 5. Repeat Steps 2 to 4 for "intelligent" choices of **exp** and choose the best fit polynomial.

To judge the goodness of fit of the polynomials, we use the *mean (average) relative error* (ARE) and the *maximum relative error* (MRE) as computed by the differences between the two delay formulae evaluated at the same points.

Note that all numerical computations required by this procedure may be performed, "simultaneously" on all the links of a given link set, by using the computer programs presented in the Appendix. As to the "educated" guesses that are also a necessary part of the procedure, the experiments, and their analysis, that we present in the rest of the section reveal much of the essence of choices made.

4.2 Numerical experiments

The sets of points (T_a, D_a) were first generated for all the links in the sample. The procedure considers all integer values for the total train flow, $T_a = t_a + t_{\bar{a}}$, between 0 and 92% of each link's capacity. For *even* total train flow values, only one delay evaluation is performed, supposing equal train flows in each direction. For *odd* values, two delays are computed: one using $t_a = (T_a + 1) / 2$ and $t_{\bar{a}} = (T_a - 1) / 2$, and another using the reverse values. The average of these two delays is then used. Figure 2 displays the plot of these points for link number 1. Note that, since link 1 is as representative as any other link in the sample and since we want to keep the presentation simple, only plots relative to link 1 are illustrated in this report.

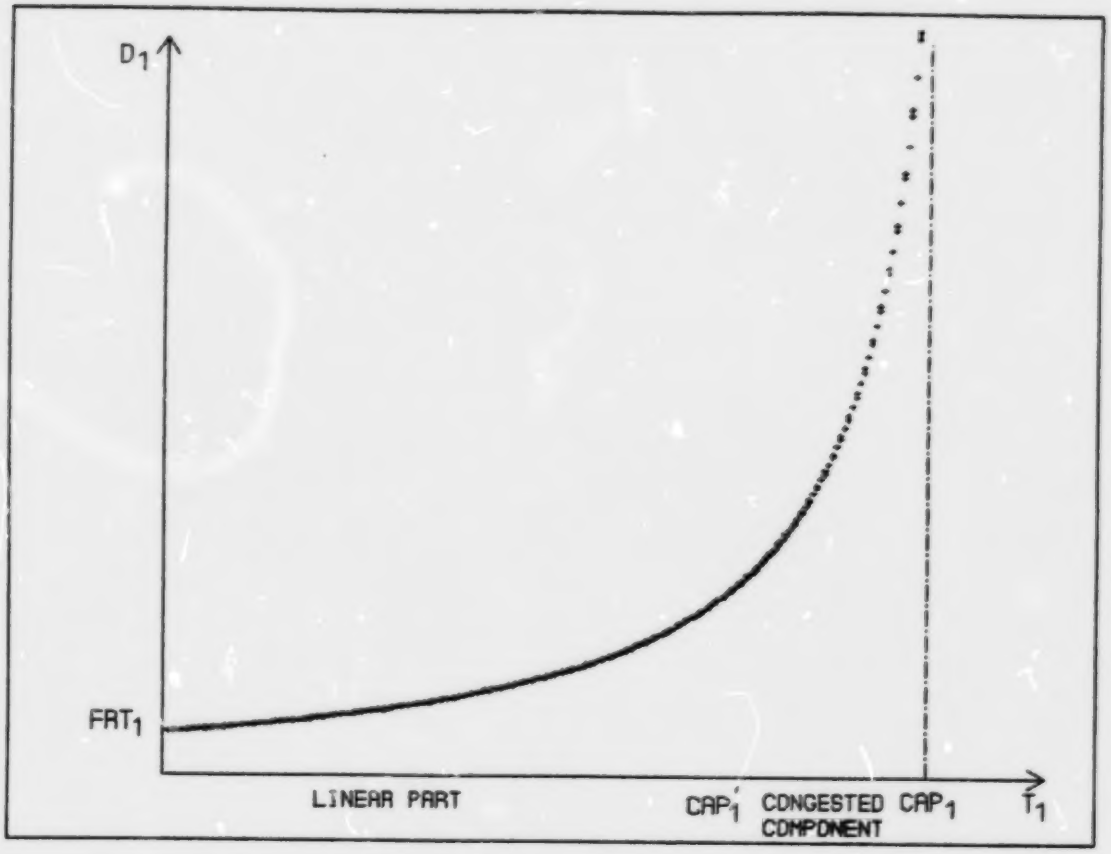

Figure 2. Plot of the points (T_a, D_a) for link 1

The choice of the degree of the polynomial, **exp**, is guided by a dual criterion. On one hand, as good a fit as possible is desired for the polynomial. This usually means, since it improves the fit in the "congested" part of the plot, a high degree. On the other hand, the inclusion of high degree polynomials in the objective function of the STAN model may induce numerical problems when a multimode assignment is performed. In particular, these problems may seriously affect the procedure that numerically computes the derivatives of the function and are especially important in high congestion situations when flows are near capacity. The convergence and precision of the algorithm may then be affected. Or, on one hand, congestion in the yards and on the lines of a rail transportation system is usually quite high, and, on the other hand, the STAN assignment algorithm requests a very large number of evaluations of both functions and theirs derivatives. One would then prefer a polynomial degree as low as possible. A compromise has to be reached.

For the links of our sample, we tried **exp** equal to 5, 6 and 7, values which seemed reasonable for the observed plots (of which Figure 2 is an example) and for an efficient assignment algorithm. The results are displayed in Table 2, while Figure 3 illustrates them for link 1.

link number	exp = 5		exp = 6		exp = 7	
	ARE	MRE	ARE	MRE	ARE	MRE
1	8,16	27,00	7,19	20,57	7,98	17,51
2	8,31	27,48	7,25	20,90	7,98	17,69
3	7,68	25,31	7,00	19,54	7,96	17,08
4	8,08	26,73	7,17	20,44	8,00	17,52
5	7,83	25,52	7,12	19,54	8,09	17,02
6	8,19	27,10	7,20	20,64	7,98	17,53
7	8,21	27,15	7,20	20,68	7,97	17,57
8	7,97	26,24	7,14	20,13	8,00	17,37
9	8,16	26,92	7,20	20,48	7,99	17,40
10	7,98	26,69	7,25	20,78	8,21	18,09

Table 2. Fitness characteristics for polynomials of various degrees

One notes that polynomials of degree 6 and 7 are significantly better than polynomials of degree 5. The difference is much less significant, however, when comparing polynomials of degree 6 and 7 only. Therefore, consistent with the previous discussion, we choose a degree 6 for our delay functions.

The proper estimation of ϕ_2 and, consequently, that of the polynomial, depends upon the right choice of CAP'_a. This value indicates, as a certain percentage of the link capacity, the threshold where the link begins to be considered congested. It is also used to indicate in the calibration procedure, the point where the two separate calibrations (i.e. of ϕ_2 in the congested part of the function and of ϕ_1 in its linear part) meet and are linked.

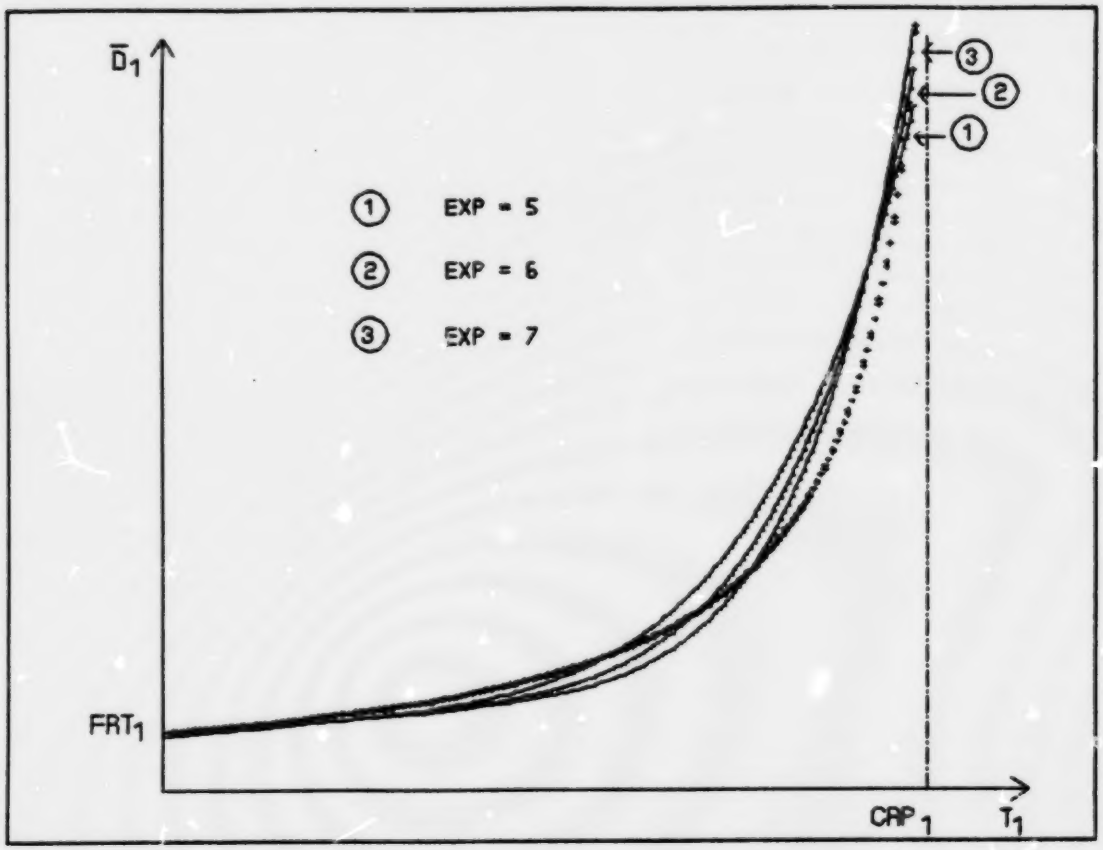

Figure 3. Polynomial delays of various degrees

CAP'_a should therefore be chosen at an "equilibrium" point between the linear and congested components of the delay function (see Figure 2). Choosing it too low would imply a poor fit of the congested part of the polynomial. On the other hand, too high a value for CAP'_a generates unsatisfactory results for the linear component of the polynomial approximation.

These observations are illustrated by the figures of Table 3 showing the *mean relative errors* and *maximum relative errors* of several polynomial approximations of the over-the-line delay for link 1, obtained with varying congestion thresholds. Figure 4 displays some of these polynomials. Consequently, a value of $CAP'_a = 80\% * CAP_a$ was deemed satisfactory for the links of our sample. Note that this 80% value for the congestion threshold is in agreement with previous results (Roy, 1986) obtained by using a different link sample.

Congestion Threshold	Rule 3'		Rule 3	
	ARE	MRE	ARE	MRE
0,40%			29,99	109,20
0,50%	44,37	159,81	20,75	75,67
0,60%	21,93	80,02	11,43	39,67
0,70%	13,18	46,76	7,90	24,22
0,76%	10,95	37,66	7,22	20,76
0,80%	10,25	34,74	7,19	20,57
0,85%	10,29	34,91	7,70	23,25

Table 3. Fitness characteristics, varying congestion thresholds and calibration rules

It is also noteworthy that, in Step 3 of the calibration procedure, an alternate rule may be used to evaluate ϕ_2:

STEP 3'. Estimate ϕ_2 by fitting

$$D_a = FRT_a * (\phi_2 * (T_a / CAP'_a)^{exp})$$

for high values of T_a, i.e. $T_a \geq CAP'_a$.

Table 3 displays the results obtained for link 1 using the two rules, 3 and 3' respectively. As can be seen, rule 3 consistently outperforms rule 3', and this observation is true for all the links of the sample. This behaviour may be explained by the fact that rule 3 makes use of existing information that the alternate rule discards: FRT_a , the "fixed cost" of traveling on link a. Or, using the "fixed cost" information in Step 3 of the procedure has the effect of decreasing the value of ϕ_2 and, consequently, the relative weight of the non-linear component of the polynomial approximation. Graphically, this corresponds to a "lowering" of all the congested area toward the linear part of the plot. Consequently, the subsequent fitting of the linear component and the linking of the two separate calibrations are improved, thus improving the overall fit of the polynomial. Consequently, we

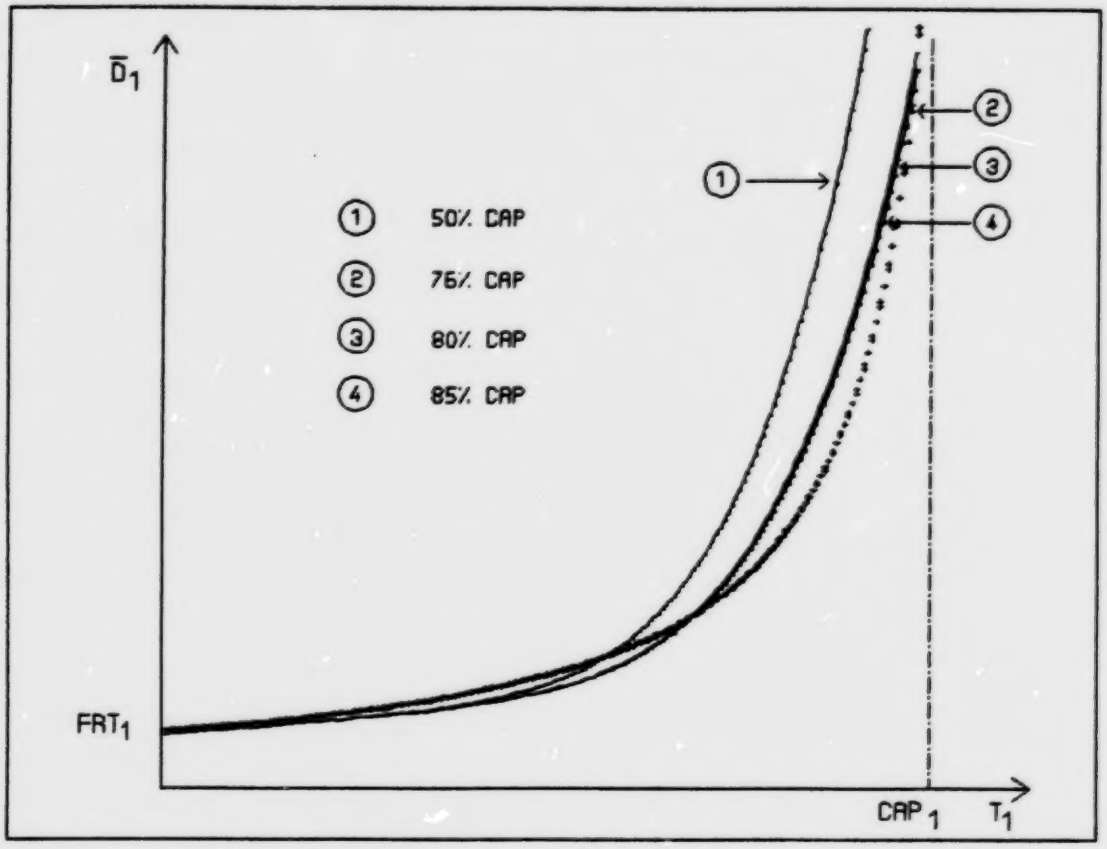

Figure 4. Polynomial delays, varying congestion thresholds

recommend the use of rule 3 in the calibration procedure.

Using $CAP'_a = 80\% * CAP_a$, $T_a^* = 0{,}1 * CAP_a$ and **exp** equal to 6, we obtained polynomial delay functions, one for each link in the sample; their parameters and corresponding goodness of fit are displayed in Table 4. These results are extremely satisfactory.

4.3 UTILIZATION MODES

The calibration procedure, used as described in the previous section, generates good polynomial approximations for the over-the-line delays for the links of an arbitrarily large network. It generates, in fact, **one** function for each link, and these functions are likely to be different for each link. For STAN and most applications of realistic dimensions, this is not acceptable. We therefore present in this section how to use the same procedure to calibrate a limited number of "good" polynomial

link number	ϕ_2	ϕ_1	1,1/CAP	ARE	MRE
1	4,127	0,00521	0,00517	7,19	20,57
2	4,143	0,00573	0,00568	7,25	20,90
3	4,079	0,01521	0,01512	7,00	19,54
4	4,122	0,00895	0,00889	7,17	20,44
5	4,096	0,02251	0,02272	7,12	19,54
6	4,130	0,00496	0,00492	7,20	20,64
7	4,131	0,00425	0,00421	7,20	20,68
8	4,108	0,00754	0,00750	7,14	20,13
9	4,124	0,00466	0,00463	7,20	20,48
10	4,163	0,04266	0,04324	7,25	20,78

Table 4. Polynomial over-the-line delays

functions for the links of a realistically large network.

First, let us examine the polynomials generated at the end of the previous section for the links of our set. Two main conclusions may be drawn from the figures displayed in Table 4:

a) ϕ_2 presents very similar values for all links;

b) Due to the chosen value for T_a^*, ϕ_1 is always very close to 1,1/CAP which would suggest that capacity is a good criterion for the grouping of links.

We may use then the same polynomial for all the links of the network. In the case of our sample data, using exp= 6, ϕ_1= 1,1/CAP and ϕ_2= 4,12252, the average of the values of ϕ_2 for all links, we obtained the results displayed in Table 5.

This method may, however, generate quite significant errors for large networks, since there may be links with very different characteristics of length, capacity, traffic, etc. It may be interesting then to group links into classes and calibrate, following the preceding method, one polynomial for each of these classes. Considering the previous remark on the value of ϕ_1 and the known relationships between

link number	ARE	MRE
1	7,236	20,341
2	7,239	20,339
3	7,192	20,330
4	7,223	20,338
5	7,135	20,229
6	7,237	20,345
7	7,239	20,344
8	7,226	20,337
9	7,239	20,340
10	6,989	20,138

Table 5. Fitness of the "average" polynomial

capacity, congestion and delay, classes should be defined according primarily to the link capacity and, secondly, to other link characteristics, such as the free running time.

For our example, four link classes may be formed:

A: large capacity, links 1,2,6,7 and 9;

B: medium capacity, links 4 and 8;

C: small capacity, links 3 and 5;

D: very small capacity, link 10.

For each class, the parameters of the polynomial delay and the corresponding fitness characteristics are shown in Table 6, where the values of ϕ_1 and ϕ_2 for each class are the averages over the values obtained for the polynomial of the links in that particular class.

This procedure gives generally better results than the previous one and is preferable, in particular when the network contains links of significantly different types.

class	exp	ϕ_1	ϕ_2	ARE	MRE
A	6	0,00492	4,12	7,625	19,495
B	6	0,00819	4,12	7,893	18,986
C	6	0.01892	4,12	6,517	24,513
D	6	0,04324	4,12	6,978	20,086

Table 6. Polynomial delays for each class

We note, finally, that all these procedures may be used even when the network is totally or partially hypothetical, as in the case of planned modifications to the infrastructre. It is sufficient to specify the corresponding values for the link characteristics, especially capacity and free running time, and the calibration procedure may be applied.

4.4 Validation - regression procedures

The parameters of the approximation polynomial for the over-the-line delay function may also be estimated directly, for a given link, by non-linear regression methods. This approach could, at least theoretically, provide us with more accurate approximations than the heuristic method. Yet, for reasons that will became clear by the end of this section, we do not retain this approach. We do, however, compare the two methods to validate the results of the heuristic calibration procedure.

Our experiments were performed using the non-linear regression program AR from the BMDP package on four links representative of their respective link class: links number 1, 4, 3 and 10.

Initially, the parameters of the four polynomials were calibrated using the regression program without any constraint. The results are presented in Table 7.

Since the degrees of the polynomials estimated by the unconstrained non-linear regression procedure are too high for our requirements, we reran the regression calibration procedure with the degree of the polynomial fixed at six. Table 8

link	exp	ϕ_2	ϕ_1	ARE	MRE
1	8,96	2,4882	0,009215	6,014	11,797
4	8,92	2,4968	0,015815	5,950	11,699
3	8,45	2,5800	0,026000	5,330	10,508
10	8,93	2,5174	0,076680	5,752	11,577

Table 7. Unconstrained non-linear regression

presents the results of this experiment, while Table 9 displays the corresponding figures for the heuristic calibration procedure. Figure 5 displays the polynomials, for link 1, that correspond to these three methods.

link	exp	ϕ_2	ϕ_1	ARE	MRE
1	6	3,56347	0,00558	5,02	14,92
4	6	3,55450	0,00965	4,94	13,66
3	6	3,42230	0,01743	4,17	12,54
10	6	3,67862	0,04442	5,64	11,22

Table 8. Non-linear regression with constrained degree

Several conclusions may be drawn from an examination of these results, which we summarize as follows:

a) The polynomial functions obtained by using non-linear regression methods, either constrained in the degree of the polynomial or not, do not offer a significantly better fit than those obtained by the heuristic calibration procedure. The *average relative errors* are similar in all cases. As for the *maximum relative errors*, they are indeed lower for the regression methods but, the MRE being an extreme mesure, they are not an adequate criterion in this case, as can be observed in the plots of Figure 5.

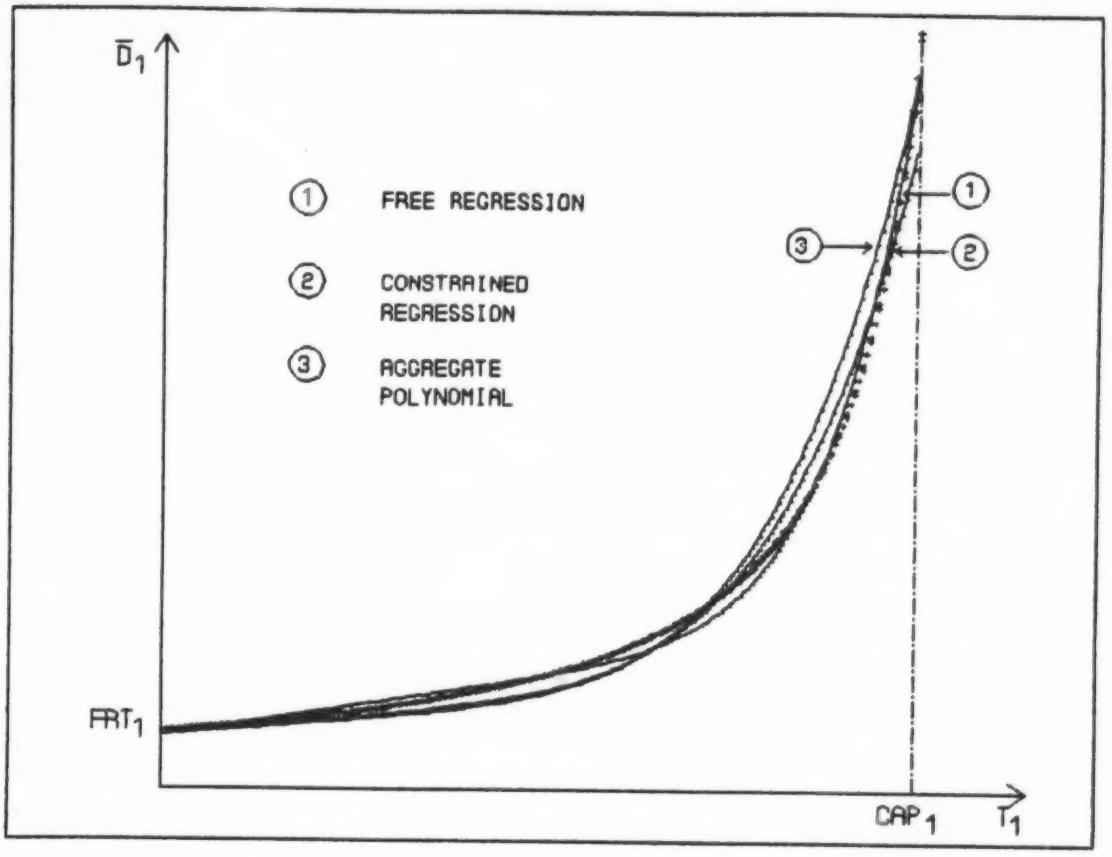

Figure 5. Aggregate and non-linear regression polynomials

link	exp	ϕ_2	ϕ_1	ARE	MRE
1	6	4,12	0,00492	7,625	19,495
4	6	4,12	0,00819	7,893	18,986
3	6	4,12	0,01892	6,517	24,513
10	6	4,12	0,04324	6,978	20,086

Table 9. Heuristic calibration procedure

b) The two non-linear regression methods result in polynomial functions of pratically similar quality, as measured by their goodness of fit.

b) The polynomial degree generated by the unconstrained regression method is generally too high for the desired efficiency of the assignement algorithm, as discussed earlier.

One also notes that the regression approach still requires the execution of Step 1 (delay data generation) as well as most of the data manipulation of our procedure. On the other hand, it offers less possibilities for user control and aggregation when a limited number of polynomials have to be selected for a large number of links.

It is thus quite clear that the non-linear regression approach is not entirely satisfactory for our function fitting problem; nevertheless, it permitted us to validate a heuristic procedure which offers extremely satisfatory results and an adequate degree of flexibility and adaptibility.

V. TRAIN AND DELAY APPROXIMATION

This section is dedicated to a brief analysis of the train flow approximation formula. We also briefly examine the behavior of the train and delay approximations when the unique train type hypothesis is relaxed.

We have seen in Crainic, Florian and Leal (1987) that there are but two methods to approximate, at the link level, the complex process that determines, at the network level, the train composition, length and frequency: an *average* formula and a *by product* approach. Given the *typical car* defined, for each product **p**, by its dead weight vw_p and its average load w_p, and given the link flows v_a^p, one approximates the *number of cars* n_a^p of product **p** travelling on a particular link **a** by

$$n_a^p = v_a^p \; / \; w_p$$

and the *average* formula gives then a total link flow t_a of

$$t_a = \sum_p n_a^p \; / \; n$$

for an average number of cars per train **n**.

The *by product* approach, used in STAN, computes the number of trains travelling on a link by

$$t_a = \sum_p \frac{v_a^p + (v_a^p \; / \; w_p) * vw_p}{\alpha_p * \beta_a}$$

where α_p represents the weight of a train made-up entirely of cars of product **p** while β_a is the link correcting factor that ajusts the approximated train flow to the particular features (maximum load, signalization, etc.) of the link.

We have already identified (Crainic, Florian and Leal, 1987) the undeniable conceptual and modeling advantages of the second formula. We now present some figures to numerically back-up our analysis.

Table 10 displays the values of the train traffic, on each direction of each link, for the transportation plan generated by the tactical model (Crainic, Ferland and Rousseau, 1984) on the network from which the sample data was extracted (the *Planned* columns) and for the two approximation formulas (the *By product* and the *Average* columns, respectivelly). Two estimations of the α_p were used for the *by*

product approximation and will be discussed momentarily. Meanwhile, we focus our attention on the figures in the *Minimum* columns. Values of $\beta_a = 1$ and $n = 100$ are assumed throughout the experiments. Note that the *Planned* entries were computed in integer arithmetic, while all other figures were computed in real arithmetic and were subsequently rounded off.

Link Number	Planned		By product				Average	
			Minimum α		Maximum α		No α	
	a	$\bar{a}$	a	$\bar{a}$	a	$\bar{a}$	a	$\bar{a}$
1	30	30	27	27	12	11	24	24
2	48	48	47	46	25	23	42	41
3	7	7	3	5	1	2	3	4
4	50	48	49	49	27	34	43	43
5	37	38	40	38	21	26	35	36
6	3	3	3	3	2	2	2	2
7	5	5	6	6	2	3	5	6
8	11	10	11	9	5	4	9	8
9	4	6	3	5	2	2	3	4
10	6	6	3	3	2	1	3	3

Table 10. Planned and estimated train link flows

An examination of these figures shows that reasonable approximations may be obtained for the train link flows by using the *average* and the *by product* formulae. It also clearly shows that the second approach systematically and significantly outperforms the *average* method.

The fit is less good on links where a relatively small amount of traffic is moved, in the tactical transportation plan, by a number of trains significantly different from the number normally suggested by the car and product flows. This situation usually occurs on secondary lines and is generated either by the particular

characteristics of the line, or by local consideration of service criteria. A judicious use of the link correcting factor β_a will allow, however, to calibrate the *by product* formula to obtain an excellent fit even in these cases. This is evidently not true of the simple average method, which again demonstrates the superiority of our approach.

Our experiments, as illustrated in the figures of Table 10, have shown that some attention should be paid to the estimation of the α_p, the *by product train weights*. The difficulty here lies in obtaining figures which give a good calibration of the train link flow formula for all the links of the network, given that each product is generally transported between several origin-destination pairs. As discussed in Crainic, Florian and Leal (1987), several products may be represented in the model by the same aggregated "real" commodity. This would allow a more "personalized" calibration of the corresponding α_p on groups of origin-destination pairs with similar loading characteristics. It would be utopic to believe, however, that one could realistically consider each origin-destination-commodity combination as a separate product. Hence, the need to calibrate the factor α_p for several origin-destinations pairs for each product.

We then examined, for each product in the sample, the distribution of its α_p factor over the origin-destination pairs between which it is transported and Table 11 displays the average and extreme values of these distributions.

These figures illustrate our observation that, for most products moved through a transportation system, the average car load for a given product presents generally small variations among the various origin-destination pairs: for most products in the sample the difference between the maximum and minimum figures is relatively small, while the average value for the car load is very close to the minimum load figure. The utilization in this case of the minimum of the distribution, instead of the mean, will generate a higher number of trains and, thus, have a correcting effect on the tendency of all approximation formulae based, even loosely, on averages, to underevaluate the actual figures.

The case of product 99 illustrates the other situation, when there are significant variations among the origin-destination pairs for the same product. For

Product	Average	Maximum	Minimum
1	119,8	150	110
5	90,0	90	90
6	129,1	140	125
7	134,1	150	125
8	130,0	130	130
13	120,0	120	120
15	108,3	140	75
16	130,0	130	130
17	74,5	80	70
18	65,1	70	60
19	70,0	70	70
20	70,0	70	70
21	108,6	130	90
22	93,6	100	85
23	94,3	100	75
24	100,0	100	100
25	87,8	100	85
26	94,4	100	95
27	80,9	100	65
28	83,5	100	75
29	28,4	40	25
30	57,3	60	50
31	78,8	90	75
32	60,4	70	60
99	56,5	150	50

Table 11. Average and extremes of the α_p distributions

(hundreds of tons)

each origin-destination pair, commodity 99 is the aggregation of all products for which the shipment size is under a certain tonnage threshold. Weighted averages were used to compute the various product characteristics. This aggregation generated large variations in the characteristics of product 99 among origin-destination pairs. For this type of product, better results are obtained if the corresponding α_p is measured where the car load figures for most origin-destination pairs are concentrated: as the median of the distribution, instead of its minimum.

We suggest therefore the following procedure to calibrate the train load factor α_p for a given product p:

1) Determine the train load factor for each origin-destination pair implicated in the transportation of the product, and consider the resulting distribution of the α_p;

2) If the loading is homogeneous, i.e. small difference between the minimum and maximum values of the distribution and the mean is close to the minimum, then fix the value of the α_p factor equal to the minimum train load for that product;

3) Else (heterogeneous loading), fix the α_p equal to the median of the distribution for the particular product.

We used this procedure to calibrate the α_p for the train approximation formula and generated the figures appearing in the columns *By product - Minimum* of Table 10. As already noted, these figures are better than those obtained by the *average* approach that corresponds to the utilization of a unique α factor for all products. Table 10 also displays the figures corresponding to the α_p fixed to the maximum values of the distributions. This option is clearly non satisfactory.

Before concluding, we briefly examine the question of the overall precision of our approximations, especially when the hypothesis of a unique freight train type and no passenger trains on the network is not verified.

Table 12 presents, for each link in our sample, the number of *normal* and *rapid* freight trains travelling in both directions on the link, as planned by the

tactical model for the initial case. It also displays the train flows estimated by our method from the product flows determined by the same transportation plan, and the estimated link capacity. Table 13 displays the corresponding over-the-line delays computed (in minutes), from the train flows of Table 12, by the tactical model, the Petersen's delay formula and our own approximated polynomial delay.

Link Number	Normal freight	Rapid freight	Total planned	Estimated capacity	Estimated T_a
1	39	21	60	213	54
2	66	30	96	194	93
3	14	0	14	73	8
4	98	0	98	124	98
5	75	0	75	48	78
6	6	0	6	224	6
7	10	0	10	261	12
8	21	0	21	147	20
9	10	0	10	234	8
10	12	0	12	25	6

Table 12. Total train link flows

Several observations may be made based on the figures displayed in tables 12 and 13. First, the results reconfirm that the polynomial delay offers a very good approximation of the delay formula proposed by Petersen. One also notes that, while Petersen's formula computes a negative delay, the delay function that we propose manages successfully the case where a train flow superior to the link capacity has been assigned to a particular link (e.g. link 5).

When comparing the polynomial and the tactical planning delay formulae, one should first note that the capacities used in our experiments were estimated by using approximate figures only. Since the capacity information was not required in the tactical model (it is implicitly defined into the coefficients of the over-the-line

Link Number	Planned rapid	Planned normal	Polynomial delay	Petersen's formula
1	339,42	415,62	371,71	391,83
2	21,96	26,94	30,78	35,78
3	-	560,40	460,11	458,87
4	-	806,94	3140,67	2669.74
5	-	702,66	133541,27	-779,34
6	-	72,90	53,39	53,28
7	-	100,98	76,18	75,55
8	-	226,98	187,23	184,17
9	-	235,68	174,54	174,78
10	-	73,44	66,33	68,77

Table 13. Planned and estimated over-the-line train delays

delay function) and, thus, was not readily available, we had to make some educated guesses. These are clearly wrong for some of the links (links 4 and 5, particularly), which biases the results for these links, at least. Yet, the overall performance of the polynomial approximation function is very satisfactory indeed.

The approximation formula generally underestimates the delay computed by the tactical planning function and relative errors between 10% and 27% are observed between the delays computed by the two functions. This is very satisfactory when one considers the cascading approximations implicit in the polynomial over-the-line delay: the local train approximation and the polynomial approximation of a delay function that, in turn, approximates the function used in the tactical model. Thus, an improvement in any of these approximations, especially a better fit of the train link flow estimation and a more precise computation of the link capacity figures, should greatly improve the overall precision of the over-the-line delay estimation.

It is also interesting to note that, as expected, the approximation performs better when the line is not too congested, since the polynomial delay "exagerates" the behaviour of the congested part of the delay curve. In this case, the presence on the network of several train types with different speeds and priorities has no significant impact on the degree of fitness of the polynomial delay function. This is a very encouraging result and augures well for further studies of the approximation of over-the-line delays in high congestion situations and when several train types are defined on the network.

CONCLUSIONS

We have presented and analysed approximation formulae for the total train flow and for the over-the-line train delay for the links of a rail network that may be used in a model for the strategic planning of a national or regional multimode multicommodity freight transportation system.

The approximations have proved to be accurate and, thus, totally adequate for strategic planning modeling purposes. We have, therefore, also proposed possible utilization modes for the various formulae and their corresponding calibration procedures.

We noted that the precision of the approximation formulae is quite dependent on the accuracy of the input data and we suggest, therefore, that particular attention should, at least, be paid to:

- the estimation of the link capacity;
- the estimation of the *by product train weights*;
- the utilization of the link correcting factor to adequately adjust the train approximation formula;
- the utilization of the most precise formula possible as starting point for the calibration of the over-the-line polynomial delay function.

This will improve the overall quality of the approximations and thus the validity of the flows simulated by the global network planning model.

ACKNOWLEDGEMENTS

We wish to thank M. Jacques Guélat and especially Professor Michael Florian for their thorough review of an earlier version of this paper. Their comments and advice have greatly contributed to improve its presentation.

BIBLIOGRAPHY

Crainic, T.G. (1982), *Un modèle de planification tactique pour le transport ferroviaire des marchandises*, Ph.D. thesis, publication # 247, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G., Ferland, J.A. and Rousseau, J.-M. (1984), *A tactical planning model for rail freight transportation*, **Transportation Science** 18 (2), 165-184.

Crainic, T.G., Florian, M. and Guélat, J. (1987a), *O.R. Tools for Strategic Planning of National Freight Systems: Modeling*, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G., Florian, M. and Guélat, J. (1987b), *O.R. Tools for Strategic Planning of National Freight Systems: Algorithms*, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G., Florian, M. and Leal, J.-E. (1987), *A Model for the Strategic Planning of National Freight Transportation by Rail,* publication # 518, Centre de recherche sur les transports, Université de Montréal.

Florian, M., Sénécal, L., Nguyen, S. and Crainic, T.C. (1986a), *A Multiproduct Multimode Freight Assignment Model for National Transportation Planning,* TIMS/ORSA Joint National Meeting, Los Angeles.

Florian, M., Crainic, T.G., Guelat, J. and Babin, A. (1986b), *STAN: an interactive graphic system for strategic planning of freight transportation at the national and regional level* publication # 495, Centre de recherche sur les transports, Université de Montreál.

Petersen, E.R. (1974), *Over-the-road transit time for a single track railway*, **Transportation Science** 8, 65-76.

Roy, M.-C. (1986), *Modélisation stratégique du transport des marchandises au Brésil: le mode ferroviaire et l'analyse de matrices de demande,* M. Sc. thesis, publication # 485, Centre de recherche sur les transports, Université de Montréal.

APPENDIX A - PROCEDURES

In this appendix, we present four procedures that may be used, either individually for evaluation purposes, or as part of the calibration procedure described in Section IV to generate polynomial delay functions that approximate delay formulas (here, Petersen's (1974)) that may be more precise but less computationally or analytically suitable.

There is a procedure for each of the following operations, required by the steps of the calibration procedure:

1. Generate a set of points (total link train flows) and compute link delays;
2. Evaluate the parameters of a polynomial delay function of specified degree and compute the corresponding link delays;
3. Compare two delay fonctions;
4. Evaluate link delays using a specified polynomial function.

The programs are written in FORTRAN and are listed in Appendix B. The procedures treat all the links of specified link sets.

INITIAL DATA FORMAT

The file LINKS contains the initial information associated with the links for which delays (polynomial or not) have to be evaluated. There is a record for each link, in the following format (all file descriptions in this Appendix display the *name* of the field in the first column, the *format* in the second column and the *description* of the field in the third):

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);
t_a	(F12.5)	Train flow from NO1 to NO2;
$t_{\bar{a}}$	(F12.5)	Train flow from NO2 to NO1;
DIST	(I5)	Link length;
NTP	(I4)	Number of passenger trains on the link.

STEP 1 - GENERATE DELAYS

This procedure, called FPETER, computes, for a set of links, the over-the-line delays for various train flow values using the Petersen delay formula.

The procedure considers all integer values for the total train flow, $T_a = t_a + t_{\bar{a}}$, between 0 and 92% of each link's capacity. For *even* total train flow values, only one delay evaluation is performed, supposing equal train flows in each direction. For *odd* values, two delays are computed: one using $t_a = (T_a + 1) / 2$ and $t_{\bar{a}} = (T_a - 1) / 2$, and another using the reverse values. The average of these two delays is later used by the "comparing" procedures of Step 3.

The input of the procedure is the file LINKS. The output of the procedure is the file PETER which contains, for each link, a number of records specifying the delay values for all considered train flows. The format of the file is as follows:

- Header record, description of link a

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);

- Empty record
- One point per record

T_a	(I15)	Total train flow;
D_a	(F15.5)	The computed delay.

- Last record for the link

  0 0 (2I15)

The file PETER is used as input by the two next procedures. It may also be used to draw graphs of delay versus total train flow for some or all links in the file. These graphs, in turn, may then be used to choose values for the exponent of the estimated polynomial delays.

It is noteworthy that, as long as the format of the input and output files are respected, another procedure may be substituted to FPETER if one wants to try a different delay formula. Yard delays could thus be approximated if the appropriate procedure is used instead of FPETER. It is of course understood that the new formula gives a delay of the same general shape (see figure 1) as the current one.

STEP 2 - EVALUATING PARAMETERS

This procedure, called FPHIS, computes the values of the parameters ϕ_1 and ϕ_2 of the polynomial delay, given a specified polynomial degree **exp**. The computation method has been described in Section IV. If desired, the procedure also produces the over-the-line delays generated using this polynomial.

The procedure accepts the file PETER (and the degree of the polynomial) as input. The output of the procedure is the file PHIS which contains the values of the parameters and, eventually, the computed delays.

The contents of the output file is controlled by the value of the logical variable *LONG*. The long form, *LONG = true*, has the following format:

- Header record, description of link **a**

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);
exp	(F10.5)	Degree of the polynomial delay;
ϕ_1	(F10.5)	Linear parameter of the polynomial;
ϕ_2	(F10.5)	Congestion parameter of the polynomial.

- Empty record
- One point per record:

T_a	(I15)	Total train flow;
D_a	(F15.5)	The delay computed using the estimated parameters.

- Last record for the link

0 0 (2I15)

In the short form, *LONG = false*, only the estimated parameters for each link (in the previous header link record format), plus the average value of ϕ_2, are written:

- Header record link1
- Header record link2
- ...
- Header record last link
- Average value of ϕ_2 (F15.5)

STEP 3 - COMPARING DELAYS

The procedure FECARTS compares a given delay formula with its polynomial approximation. Specifically, it computes, for all links, the average relative error (ARE) and the maximum relative error (MRL) of the polynomial delay as compared to the initial formula.

The procedure FECARTS accepts as input the files PETER and PHIS (in the *long* format) and outputs the errors by link and, eventually, by total train traffic value.

The contents of the output file, called ECARTS, is again controlled by a logical variable. The short version, *LONG = false*, gives, for each link, the mean and relative errors, in the following format:

- Header record: see the file PHIS;
- Error record:

  ARL, MRL (2F15.5).

The long version gives, in addition, the relative error for each point:

- Header record, description of link a

NO1	(I4)	origin node;
NO2	(I4)	destination node;
FRT	(F10.5)	free running time (in minutes);

CAP	(F10.5)	capacity (weekly);
exp	(F10.5)	degree of the polynomial delay;
ϕ_1	(F10.5)	linear parameter of the polynomial;
ϕ_2	(F10.5)	congestion parameter of the polynomial.

- One point per record:

T_a	(I4)	total train flow;
RE	(F15.5)	relative error for that train flow.

- Last record for the link:

ARL, MRL (2F15.5).

STEP 4 - EVALUATING POLYNOMIAL DELAYS

The procedure FPOLY computes, for all specified links, the polynomial over-the-line delay.

The procedure accepts the file LINKS and values for **exp**, ϕ_1 and ϕ_2 as input, and outputs, for each link, the delay computed with the specified polynomial.

The output of the procedure is the file POLY, that has exactly the same structure and format as the (long) file PHIS generated by the procedure FPHIS in Step 2. Consequently, the output of FPOLY may be given as input to FECARTS to evaluate the accurateness of any given polynomial.

APPENDIX B - PROGRAMS

```
      PROGRAM FPETER ( INPUT, OUTPUT, LINKS, PETER,
     +                                 TAPE5 = INPUT,
     +                                 TAPE6 = OUTPUT,
     +                                 TAPE7 = LINKS,
     +                                 TAPE8 = PETER  )
C
C     ***************************************************************************
C                                                                               *
C     INTRODUCTION:                                                             *
C                                                                               *
C                                                                               *
C        THIS PROGRAM COMPUTES A SERIES OF POINTS FOR EACH LINK                 *
C     CONTAINED IN THE FILE "LINKS".  THE POINTS REPRESENT PETERSEN'S           *
C     DELAY AS A FUNCTION OF THE NUMBER OF TRAINS / WEEK ON THE LINK.           *
C     THE NUMBER OF TRAINS "TA" VARIES BETWEEN ZERO AND A CERTAIN               *
C     PERCENTAGE OF THE TOTAL CAPACITY (VALUE OF XSTOP).                        *
C                                                                               *
C     ***************************************************************************
C                                                                               *
C     DESCRIPTION OF VARIABLES:                                                 *
C                                                                               *
C                                                                               *
C     NO1, NO2: ORIGINE AND DESTINATION NODES OF A LINK                         *
C     FRT: FREE RUNNING TIME ON CURRENT LINK (MINUTES)                          *
C     CAP: CAPACITY OF CURRENT LINK (NUMBER OF TRAINS PER WEEK)                 *
C     DPETER: DELAY COMPUTED ACCORDING TO PETERSEN'S FORMULA                    *
C                                                                               *
C     TA: TOTAL NUMBER OF TRAINS IN BOTH DIRECTIONS OF A LINK                   *
C     XSTOP: FRACTION USED TO DETERMINE 'TAMAX' FOR EACH LINK                   *
C     TAMAX: MAXIMUM NUMBER OF TRAINS WHICH IS USED TO COMPUTE A DELAY          *
C                                                                               *
C     ***************************************************************************
C
C     DECLARATION OF VARIABLES.....................................
      INTEGER  NO1, NO2, TA, TAMAX
      REAL  FRT, CAP, DELAY, XSTOP
C
C     FORMATS......................................................
C
 1001 FORMAT (2I4,2F10.5)
 8002 FORMAT (A3)
 8003 FORMAT (I15,F15.5)
 8004 FORMAT (2I15)
C
C     INITIALISATION...............................................
C
      REWIND7
      REWIND8
C
      XSTOP = 0.92
C
```

```
*     START.....................................................
*
100   READ ( 7, 1001, END=99 ) NO1, NO2, FRT, CAP
      WRITE ( 8, 1001 ) NO1, NO2, FRT, CAP
      WRITE ( 8, 8002 ) '   '
      TAMAX = INT ( CAP * XSTOP )
*
      DO 10 TA = 0, TAMAX
*                                   --> IF TA IS EVEN...
      IF (MOD(TA,2).EQ.0) THEN
         DELAY = (FRT * CAP) / (CAP - TA)
         WRITE ( 8, 8003 ) TA, DELAY
      ELSE
*                                   --> IF TA IS ODD...
         TA1 = ( TA - 1 ) / 2
         TA2 = ( TA + 1 ) / 2
         DELAY = (FRT * (CAP + TA2 - TA1)) / (CAP - TA)
         WRITE ( 8, 8003 ) TA, DELAY
         DELAY = (FRT * (CAP + TA1 - TA2)) / (CAP - TA)
         WRITE ( 8, 8003 ) TA, DELAY
      ENDIF
10    CONTINUE
      WRITE ( 8, 8004 ) 0, 0
      GOTO 100
99    CONTINUE
      STOP
      END
```

```
      PROGRAM FPHIS ( INPUT, OUTPUT, PETER, PHIS,
     +                               TAPE5 = INPUT,
     +                               TAPE6 = OUTPUT,
     +                               TAPE7 = PETER,
     +                               TAPE8 = PHIS    )
C
C     ******************************************************************
C                                                                       *
C     INTRODUCTION:                                                     *
C                                                                       *
C                                                                       *
C        THIS PROGRAM USES THE RESULTS OF "FPETER" TO ESTIMATE "PHI1"   *
C     AND "PHI2", THE PARAMETERS OF THE POLYNOMIAL:                     *
C                                                                       *
C           FRT * (1 + PHI1*TA + PHI2*( TA/CAP' )**EXPO)                *
C                                                                       *
C        THE PROGRAM COMPUTES THE PARAMETERS OF THE POLYNOMIAL          *
C     OF DEGREE "EXPO" FOR EACH LINK OF THE FILE "LINKS".               *
C     THEN, IF "LONG" IS SET AT TRUE, IT COMPUTES THE RESULTING         *
C     DELAYS . OTHERWISE, IT MERELY COMPUTES THE AVERAGE "PHI2".        *
C                                                                       *
C     ******************************************************************
C                                                                       *
C     DESCRIPTION OF VARIABLES:                                         *
C                                                                       *
C                                                                       *
C     NO1, NO2: ORIGIN AND DESTINATION NODES OF A LINK                  *
C     FRT: FREE RUNNING TIME ON CURRENT LINK (MINUTES)                  *
C     CAP: CAPACITY OF CURRENT LINK (NUMBER OF TRAINS PER WEEK)         *
C     DPETER: DELAY COMPUTED ACCORDING TO PETERSEN'S FORMULA            *
C                                                                       *
C     DPOLY: DELAY COMPUTED ACCORDING TO THE CHOSEN POLYNOMIAL          *
C     EXPO: VALUE OF THE EXPONENT IN THE POLYNOMIAL                     *
C     PHI1, PHI2: PARAMETERS WHICH DEFINE A POLYNOMIAL                  *
C     FUNCTIO: NUMBER REPRESENTING THE FUNCTION CHOSEN TO ESTIMATE PHI2 *
C                                                                       *
C     TA: TOTAL NUMBER OF TRAINS IN BOTH DIRECTIONS OF A LINK           *
C     TA1: NUMBER OF TRAINS GOING FROM 'NO1' TO 'NO2'                   *
C     TA2: NUMBER OF TRAINS GOING FROM 'NO2' TO 'NO1'                   *
C     TAMAX: MAXIMUM NUMBER OF TRAINS WHICH IS USED TO COMPUTE A DELAY  *
C                                                                       *
C     TALINEA: NUMBER OF TRAINS FOR WHICH 'DPETER' IS STILL LINEAR      *
C     XLINEAR: FRACTION USED TO COMPUTE TALINEA                         *
C     DLINEAR: VALUE OF THE DELAY COMPUTED WITH 'TALINEA' (MAY BE AN    *
C              AVERAGE IF 'TALINEA' IS ODD)                             *
C     XNONLIN: FRACTION USED TO DETERMINE WHEN A LINK BECOMES CONGESTED *
C     CAPPRIM: NUMBER OF TRAINS FOR WHICH A LINK IS CONGESTED           *
C              CAPPRIM = CAP * XNONLIN                                  *
C     XSTOP: FRACTION USED TO DETERMINE 'TAMAX' FOR EACH LINK           *
C                                                                       *
C     NB, SUM: VARIABLES USED TO COMPUTE DLINEAR                        *
C     PHI2NB, PHI2SUM: USED TO COMPUTE THE AVERAGE ESTIMATED 'PHI2'     *
```

```
*     LONG: LOGICAL VARIABLE WHICH DETERMINES THE OUTPUT OF THE PROGRAM *
*                                                                       *
*                          $$$$$$$$$$$$$$$$$$$$$$$$$                    *
*                                                                       *
*        CHECK THE VALUES OF THE FOLLOWING VARIABLES BEFORE RUNNING THE *
*     PROGRAM:                                                          *
*              EXPO (ESTIMATED FROM PETERSEN'S GRAPH),                  *
*              XNONLIN (BETWEEN 0.5 AND 0.8),                           *
*              FUNCTIO (1 OR 2),                                        *
*              LONG (TRUE OR FALSE),                                    *
*              XLINEAR ( 0.1 ), XSTOP (MUST BE THE SAME AS IN FPETER)   *
*                                                                       *
*                          $$$$$$$$$$$$$$$$$$$$$$$$$                    *
*     *****************************************************************
*
*     DECLARATION OF VARIABLES.....................................
*
      INTEGER  NO1, NO2, TA, TAMAX, TALINEA, NB, PHI2NB, FUNCTIO
      REAL  FRT, CAP, DPETER, DPOLY, EXPO, PHI1, PHI2,
     +      DLINEAR, SUM, PHI2SUM, CAPPRIM,
     +      XLINEAR, XNONLIN, XSTOP
      CHARACTER  BLANCS*3
      LOGICAL  LONG
*
*     FORMATS......................................................
*
7001  FORMAT (2I4,2F10.5)
1002  FORMAT (A3)
1003  FORMAT (I15,F15.5)
8001  FORMAT (2I4,6F10.5)
8004  FORMAT (2I15)
8005  FORMAT (F15.5)
*
*     INITIALISATION...............................................
*
      REWIND7
      REWIND8
*
      EXPO = 6
      FUNCTIO = 2
      LONG = .FALSE.
      XLINEAR = 0.10
      XNONLIN = 0.80
      XSTOP = 0.92
*
      PHI2SUM = 0
      PHI2NB = 0
*
*     START........................................................
*
100   READ ( 7, 7001, END=99 ) NO1, NO2, FRT, CAP
      READ ( 7, 1002 ) BLANCS
*
*     ESTIMATION OF PHI1...........................................
*
```

```
      CAPPRIM = CAP * XNONLIN
      DLINEAR = 0
      NB = 0
      TALINEA = INT ( CAP * XLINEAR )
      TAMAX = INT (CAP * XNONLIN * 1.5)
      DO 10 I = 0,TAMAX
      READ ( 7, 1003 ) TA, DPETER
      IF ( TA.EQ.TALINEA ) THEN
         DLINEAR = DLINEAR + DPETER
         NB = NB + 1
      ENDIF
10    CONTINUE
*
      DLINEAR = DLINEAR / NB
      NB = 0
      SUM = 0
*
*     ESTIMATION OF PHI2.....................................................
*
20    IF ( FUNCTIO.EQ.1 ) THEN
         PHI2 = ( DPETER/FRT )
     +                    / ((TA / CAPPRIM) ** EXPO)
      ENDIF
      IF ( FUNCTIO.EQ.2 ) THEN
         PHI2 = ( DPETER/FRT - 1 )
     +                    / ((TA / CAPPRIM) ** EXPO)
      ENDIF
      SUM = SUM + PHI2
      NB = NB + 1
      READ ( 7, 1003 ) TA, DPETER
      IF ( TA.NE.0 ) THEN
         GOTO 20
      ENDIF
*
      PHI2 = SUM / NB
      PHI1 = ( DLINEAR/FRT - 1 - PHI2*((TALINEA / CAPPRIM)**EXPO))
     +       / TALINEA
*
*     HEADER.................................................................
*
      WRITE ( 8, 8001 ) NO1, NO2, FRT, CAP,
     +                  EXPO, PHI2, PHI1, (1.1/CAP)
      WRITE ( 8, 1002 ) '    '
*
*     COMPUTATION OF POINTS..................................................
*
      IF ( LONG ) THEN
         TAMAX = INT ( CAP * XSTOP )
         DO 30 TA = 0, TAMAX
         DPOLY = FRT * (1 + PHI1*TA + PHI2*(TA / CAPPRIM)**EXPO)
         WRITE ( 8, 1003 ) TA, DPOLY
30       CONTINUE
         WRITE ( 8, 8004 ) 0,0
      ELSE
         PHI2SUM = PHI2SUM + PHI2
         PHI2NB = PHI2NB + 1
```

```
      ENDIF
      GOTO 100
99    CONTINUE
      IF (.NOT.LONG) THEN
         PHI2SUM = PHI2SUM / PHI2NB
         WRITE ( 8, 8005 ) PHI2SUM
      ENDIF
      STOP
      END
```

```
      PROGRAM FECARTS ( INPUT, OUTPUT, PETER, PHIS, ECARTS,
     +                                         TAPE5 = INPUT,
     +                                         TAPE6 = OUTPUT,
     +                                         TAPE7 = PETER,
     +                                         TAPE8 = PHIS,
     +                                         TAPE9 = ECARTS  )
*
*     ***************************************************************
*                                                                     *
*     INTRODUCTION:                                                   *
*                                                                     *
*         THIS PROGRAM COMPARES THE FILES "PETER" AND "PHIS". IT      *
*      COMPUTES THE RELATIVE ERROR AT EACH POINT "TA" AND FROM THESE  *
*      THE AVERAGE AND MAXIMUM ERRORS ARE FOUND FOR EACH LINK.        *
*                                                                     *
*                                                                     *
*     INPUT: THE FILES "PETER" AND "PHIS" ( TO COMPARE THE FILES      *
*             "PETER" AND "POLY" SIMPLY RENAME "POLY" ).              *
*                                                                     *
*     OUTPUT: THE FILE "ECARTS" WHICH DOES NOT LIST THE RELATIVE      *
*              ERRORS FOR EACH "TA" UNLESS THE VARIABLE "LONG"        *
*              IS INITIALIZED AT "TRUE".                              *
*                                                                     *
*     FOR A DESCRIPTION OF VARIABLES SEE FPHIS.                       *
*                                                                     *
*     ***************************************************************
*
*     DECLARATION OF VARIABLES.....................................
*
      INTEGER  NO1, NO2, TAPETER, TAPOLY, I, J, POINTM
      REAL  EXPO, FRT, CAP, DPETER, DPOLY, DMOYEN,
     +      PHI1, PHI2,APHI1,SOM, SOMME,
     +      RLERROR, MRL, ARL
      CHARACTER  BLANCS*3
      LOGICAL  LONG
*
*     FORMATS......................................................
*
7001  FORMAT (2I4,2F10.5)
8001  FORMAT (2I4,5F10.5)
1002  FORMAT (A3)
1003  FORMAT (I15,F15.5)
9001  FORMAT (I4,F15.5)
9002  FORMAT (2F15.5)
*
*     INITIALISATION...............................................
*
      REWIND7
      REWIND8
      REWIND9
*
      LONG = .FALSE.
*
*     START........................................................
```

```
*
100   READ ( 7, 7001, END=99 ) NO1, NO2, FRT, CAP
      READ ( 7, 1002, END=99 ) BLANCS
      READ ( 8, 8001, END=99 ) NO1, NO2, FRT, CAP,
     +                         EXPO, PHI2, PHI1
      READ ( 8, 1002, END=99 ) BLANCS
*
      WRITE ( 9, 8001 ) NO1, NO2, FRT, CAP,
     +                  EXPO, PHI2, PHI1
      WRITE ( 9, 1002 ) BLANCS
*
*     INITIALISATION FOR  EACH LINK.....................................
*
      TAPETER = 1
      TAPOLY = 1
      DPETER = 0
      DPOLY  = 0
      DMOYEN = 0
      SOM    = 0
      SOMME  = 0
      I = -1
      J = 0
      RLERROR = 0
      MRL = 0
      ARL = 0
*
*     ERRORS COMPUTATIONS.............................................
*
50    READ ( 8, 1003 ) TAPOLY, DPOLY
      IF ( DPOLY.EQ.0 ) GOTO 75
*
10    SOM = SOM + DPETER
      I = I + 1
      READ ( 7, 1003 ) TAPETER, DPETER
      IF ( TAPOLY.EQ.TAPETER ) GOTO 10
*
      DMOYEN = SOM / I
      RLERROR = ABS ( DPOLY - DMOYEN ) / DMOYEN
      IF ( LONG ) WRITE (9,9001) TAPOLY, RLERROR
*
      SOMME = SOMME + RLERROR
      J = J + 1
      SOM = 0
      I = 0
      IF ( RLERROR.GT.MRL ) MRL = RLERROR
      GOTO 50
*
75    ARL = SOMME / J
      WRITE ( 9, 9002 ) ARL, MRL
      WRITE ( 9, 1002 ) BLANCS
      GOTO 100
*
99    CONTINUE
      STOP
      END
```

```
      PROGRAM FPOLY ( INPUT, OUTPUT, LINKS, POLY,
     +                               TAPE5 = INPUT,
     +                               TAPE6 = OUTPUT,
     +                               TAPE7 = LINKS,
     +                               TAPE8 = POLY  )
C
C     ****************************************************************
C                                                                      *
C     INTRODUCTION:                                                    *
C                                                                      *
C                                                                      *
C        THIS PROGRAM GENERATES POINTS FOR A FIXED POLYNOMIAL.         *
C     ALL ONE HAS TO DO IS FIX ITS THREE PARAMETERS: EXPO,             *
C     PHI1 AND PHI2.                                                   *
C                                                                      *
C     INPUT: THE FILE "LINKS"                                          *
C                                                                      *
C     OUTPUT: THE FILE "POLY" WHICH HAS THE SAME FORMAT AS THE         *
C             LONG VERSION OF "PHIS" ( THE OUTPUT OF "FPHIS" ).        *
C                                                                      *
C     FOR DESCRIPTION OF VARIABLES SEE PROGRAMS FPETER AND FPHIS.      *
C                                                                      *
C     ****************************************************************
C
C     DECLARATION OF VARIABLES.....................................
C
      INTEGER  NO1, NO2, TA, TAMAX
      REAL  FRT,CAP,DPOLY,PHI1,PHI2, EXPO,CAPPRIM,XNONLIN,XSTOP
C
C     FORMATS......................................................
C
7001  FORMAT (2I4,2F10.5)
8001  FORMAT (2I4,5F10.5)
8002  FORMAT (A3)
8003  FORMAT (I15,F15.5)
8004  FORMAT (2I15)
C
C     INITIALISATION...............................................
C
      REWIND7
      REWIND8
      EXPO = 8.96
      PHI2 = 2.4882
      XNONLIN = 0.8
      XSTOP = 0.92
C
C     START........................................................
C
100   READ ( 7, 7001, END=99 ) NO1, NO2, FRT, CAP
      PHI1 = 0.009215
      WRITE ( 8, 8001 ) NO1, NO2, FRT, CAP, EXPO, PHI2, PHI1
      WRITE ( 8, 8002 ) '   '
```

```
*
*       COMPUTATION OF POINTS.....................................
*
        TAMAX = INT ( CAP * XSTOP )
        CAPPRIM = CAP * XNONLIN
        DO 30 TA = 0, TAMAX
        DPOLY = FRT * (1 + PHI1*TA + PHI2*(TA / CAPPRIM)**EXPO)
        WRITE ( 8, 8003 ) TA, DPOLY
30      CONTINUE
        WRITE ( 8, 8004 ) 0,0
        GOTO 100
99      CONTINUE
        STOP
        END
```